POST-CORRELATION INTERPOLATION FOR DELAY LOCKED LOOPS

Background of the Invention

Field of the Invention

The present invention relates generally to signal processing, and specifically to signal processing in which symbol samples for delay locked loop processing are extracted pre and post correlator processing.

Description of Related Art

In a wideband CDMA system, multiple paths of a transmitted signal can be tracked using a rake receiver implemented at a base station due to the nature of spread spectrum sequences in a wireless channel. The rake receiver functions as several receivers operating with associated propagation delays based on the respective delays experienced by arriving multipath signals. The rake receiver has a tracking loop for each rake finger of a user that tracks the movement of the fingers typically through the use of a delay locked loop (DLL). The DLL, which computes the error in, and subsequently shifts, the sample time, needs to be performed on a per finger, or multipath, basis.

To compute the error in the sample time, the DLL requires early and late sample times. Specifically, the DLL takes the early and late samples for a given multipath signal being tracked and calculates the energy for the early and late correlations. The difference between these two energies is multiplied by a pre-computed gain factor to obtain a timing correction. A third ontime sample sequence is used for demodulation purposes as it has the largest signal to noise ratio.

In conventional DLLs, interpolators provide the early, late and ontime samples so that three samples are provided for each CDMA symbol. A conventional hardware implementation often processes the early and late correlation outputs in parallel. As the

data resulting from the early and late correlations is processed using dedicated links, transfer bandwidth is not a problem.

In new high end devices, more processing is being moved to digital signal processors. In addition, the potential for performance improvements via more sophisticated software algorithms creates motivation to place more functions like DLL processing in software. However, this shift in processing puts a strain on DSP bandwidth, as such bandwidth is typically small enough to create processing bottlenecks at the DSP I/O interface and therefore slow down DSP processing times.

Clearly a need exists for more efficient signal processing systems for, for example, delay locked loops.

Brief Description of the Drawings

Objects and advantages of the present invention will be more readily apparent from the following detailed description of preferred embodiments thereof when taken together with the accompanying drawings in which:

FIG. 1 is a diagram of an exemplary stream of interpolated CDMA symbol samples of the type produced by a preferred embodiment of a post-correlation processing system according to the present invention;

FIG. 2 is a block diagram of a preferred embodiment of a post-correlation processing system according to the present invention; and

FIG. 3 is a block diagram of a rake receiver using the post-correlation processing system shown in FIG. 2.

Detailed Description of the Preferred Embodiments

Referring now to the drawings in which like numerals reference like parts, FIG. 1 shows generally at 10 an exemplary stream of data symbol samples, such as an interpolated sequence of CDMA data symbol samples, which, as is well known in the art, are digital representations of transmitted analog signal components. These symbol samples include early (E), ontime (O), and late (L) symbol samples. The early and late symbol samples E, L are generated from out of phase chip samples and therefore have smaller signal to noise ratios than the ontime symbol samples O due, for example, to interference from other signals. All of the samples E, O, L are of the type produced by a preferred embodiment of an inventive signal processing system, specifically a post-correlation processing system 12 (FIG. 2) which will now be discussed in detail.

Referring to FIG. 2, the post-correlation processing system 12 performs an essential function for delay locked loops and processing therein and includes an I/O interface 14 for receiving input signal samples, preferably, CDMA chip samples such as those generated at a known analog processing circuit (FIG. 3) that samples received signals, preferably, CDMA received signals at, for example, two times the chip rate. The I/O interface 14 then distributes identical sets of these samples or input samples or chip samples to an interpolator 16, a first multiplexer 18, and a second multiplexer 20. Specifically, the I/O interface 14 distributes the chip samples directly to the interpolator through an interpolator input 16a and also directly to the first and second multiplexers 18, 20 through first and second multiplexer inputs 18a, 20a, respectively. The interpolator 16 is known and introduces a time offset to the input chip samples and interpolates between the input chip samples to increase chip resolution from, for example, 1/2 chip resolution if the chip samples are sampled at two times the chip rate to, for example, 1/8 chip resolution. The interpolator 16 thereby recovers time shifted samples or chip samples that are shifted in time relative to the chip samples from the I/O interface 14. Note that if or when the time reference (not shown) used to generate the samples or chip samples from I/O interface 14 is in phase or synchronized or nearly so with a desired CDMA spreading sequence these chip samples from I/O interface 14 will be "ontime" samples or "ontime" chip samples and the time shifted samples or time shifted chip samples recovered by

interpolator 16 will then be non-ontime samples or non-ontime chip samples according to the conventions used within this disclosure. It follows that non-ontime is a general reference to either late or early samples or chip samples that are not in phase with any CDMA spreading sequences for any users/subscribers. In any event the recovered time shifted (whether non-ontime or ontime) chip samples are then distributed to both the first and the second multiplexers 18, 20 through multiplexer inputs 18b, 20b.

It should be noted that the chip samples input directly from the I/O interface 14 into the first and second multiplexers 18, 20 on the multiplexer inputs 18a, 20a and the time shifted chip samples generated by the interpolator 16 and input into the first and second multiplexers 18, 20 through the multiplexer inputs 18b, 20b enable the first multiplexer 18 to always output an ontime correlator signal, specifically ontime chip samples and the second multiplexer 20 to always output a non-ontime correlator signal, specifically non-ontime chip samples. In summary this is accomplished by using an error signal (not shown) provided according to known DLL processing procedures that is indicative of whether actual sampling times are equal to desired sampling times, specifically whether the sampling clock is properly synchronized with the desired received signal or here the desired received CDMA spreading sequence. This error signal is used to adjust, retard or advance, the sampling clock and also used to control or to generate signals that control the multiplexers 18, 20 to select or provide, respectively, the ontime and non-ontime samples.

By way of example, suppose the chip samples input into the first multiplexer 18 and the interpolator 16 from the I/O interface 14 are ontime chip samples, the interpolator 16 would generate non-ontime chip samples as discussed above. However, if the chip samples input into the first multiplexer 18 through the multiplexer input 18a and into the interpolator 16 through the interpolator input 16a are non-ontime chip samples, such as early or late chip samples E, L, the interpolator 16 would perform the necessary interpolation operation to time shift the non-ontime chip samples to generate ontime chip samples for output to the first multiplexer 18 through the multiplexer input 18b. Similarly, the second multiplexer 20 will always output non-ontime chip samples, as it is controlled to select the non-ontime chip samples from the chip samples input directly from the I/O

interface 14 on the multiplexer input 20a and the interpolated and time shifted chip samples input from the interpolator 16 on the multiplexer input 20b.

These input samples and time shifted samples are selectively coupled by the multiplexers 18, 20 to correlators 22, 24, 26. Specifically identical sets of ontime chip samples output from the first multiplexer 18 are input into first and second correlators 22, 24 through first and second correlator inputs 22a, 24a, respectively, while the non-ontime chip samples output from the second multiplexer 20 are input into a third correlator 26 through a third correlator input 26a. Generally, the first, second and third correlators 22, 24, 26 strip away, or de-spread, the CDMA fast sequences from the respective input chip samples and then sample the de-spread chip samples at a symbol rate to produce symbol samples, or digital representations of components of a particular analog wireless communications signal, as is well known in the art. Generally as is known the first and third correlators 22, 26 correlate using a control spreading sequence and the second correlator correlates using a data spreading sequence a is known. By acting as a lowpass filter and by producing enough symbol samples within a specified time period, the correlators 22, 24, 26 are each capable of isolating, in the digital domain, the waveform representing the particular wireless communications signal and then outputting the appropriate symbol samples for use in post-correlation processing in accordance with the present invention.

More specifically, the first and second correlators 22, 24 extract or generate ontime control (used for timing control, format information, and the like as known) and data symbol samples O, O_d respectively from the ontime chip samples input through the correlator inputs 22a, 24a. The ontime control symbol samples O generated by the first correlator 22 are input to a post-correlation interpolator 28, part of a post correlator or post correlation processor, preferably DSP based, through a post-correlation interpolator input 28a, while the ontime data symbol samples O_d generated by the second correlator 24 are available for further processing for purposes of communicating voice or data to the user. The first non-ontime symbol samples, specifically non-ontime control symbol samples, which, as already discussed, may be either early or late symbol samples E, L, generated by the third correlator 26, are also input along with the ontime control symbol samples O as a

postcorrelator data stream into the post-correlation interpolator 28 through the post-correlation interpolator input 28a. Combining the symbol stream from correlator 22 and the symbol stream from correlator 26 provides a symbol stream at two times the symbol rate or the Nyquist rate for the control symbol samples as required for use by the post-correlation interpolator 28.

It should be noted that the correlators 22, 24, 26 process or operate on samples or chip samples at the chip rate rather than two times the chip rate. Thus the two times chip rate or Nyquist rate chip samples input to the interface I/O that are required at the input 16a for the interpolator 16 to function properly will have to be reduced in sample rate to chip samples at the chip rate at some point in the processing chain. This can be done by discarding every other sample or potentially by some known combinatorial process to provide one sample from every two at either; the input sections of each of the correlators 22, 24, 26, or within the multiplexers 18, 20, or at the output of the interface I/O 18a, 20a and the output of the interpolator 16. This selection is largely a matter of choice, however since it is usually more efficient to operate at slower speeds some may have a preference to do this as early in the processing chain as possible.

The post-correlation interpolator 28, which as is known may be either a hardware or a software implemented interpolator (software running on a DSP for example), processes the input ontime control and non-ontime control symbol samples along with other ontime control and non-ontime control symbol samples from the earlier and later time periods surrounding the ontime control and first non-ontime control symbol samples to produce second non-ontime symbol samples that are either the early or late symbol samples depending on the non-ontime symbol samples input into the post-correlation interpolator 28. Specifically, if the post-correlation processing system 12 is configured so that early symbol samples E are input into the postcorrelator processor, specifically post correlation interpolator 28, the postcorrelator interpolator 28 will generate late symbol samples L. If, however, the post-correlation processing system 12 is configured so that late symbol samples L are input into the postcorrelator processor, the post-correlation interpolator 28 will generate early symbol samples E.

The post correlation interpolator 28 is capable of generating these second non-ontime symbol samples because it includes a memory (not shown) for storing symbol samples from previous symbol sampling periods. The post correlation interpolator 28 combines symbol samples from these stored symbol samples with the ontime control and first non-ontime symbol samples input on the postcorrelator processor input 28a, weights the input symbol samples and the stored symbol samples based on their timing (early, late, ontime, sampling offset) and the lowpass filter shape arising from the processing chain, and generates the resulting second non-ontime symbol samples at the symbol sampling rate that is much slower, and requires less processing power, than the chip sampling rate necessary at the interpolator 16. The resulting second non-ontime symbol samples are then output along with the first non-ontime symbol samples originally input into the postcorrelator processor for delay locked loop processing, while the ontime control symbol samples are output for control channel processing.

It should be appreciated that the post-correlation processing system according to the above discussed preferred embodiment of the present invention reduces processing requirements, and therefore hardware complexity and cost, associated with processing wide band spread spectrum signals such as CDMA chip samples prior to DLL processing. More particularly, the post-correlation processing system 12 reduces the number of precorrelator processing interpolators from two to one, as extraction of either early or late symbol samples is performed at a post-correlation symbol sampling rate that is less intensive from a processing standpoint than interpolation processing at a pre-correlation chip sampling rate. However, the post-correlation processing system 12 may reduce the number of precorrelator processing interpolators from three to one, as the interpolator 16 is usually hardware implemented and may require interpolation for early, late and ontime chip samples if the input chip samples are not the samples that are needed for interpolation. If the post-correlation interpolator 28 is software implemented, it may change its filter coefficients to facilitate receipt of the necessary symbol samples.

In addition, because the extraction of either early or late symbol samples is executed in a post-correlation manner, the number of input lines into the DSP, and therefore the DSP control data rate, is reduced. This reduction in the DSP control data rate

frees up additional DSP I/O bandwidth and therefore greatly reduces the chance of a processing bottleneck occurring at the DSP I/O interface. Further, the post-correlation processing system 12 reduces the number of operations required to perform correlation functions by almost 1/2 because it replaces a chip rate interpolator with a symbol rate interpolator, and it eliminates a correlator that would otherwise be required for the above-discussed second non-ontime samples.

FIG. 3 shows, in block diagram form, a preferred embodiment of an exemplary implementation of the post-correlation processing system in a rake receiver 30, such as may be utilized in a CDMA receiver for a cellular base station. However, it should be appreciated that this specific implementation is exemplary only, as the post-correlation processing system of the present invention may be implemented in any type of environment in which correlation functions must be processed, such as, for example, fast hopped spread spectrum communications systems as well as mobile handsets or radios within a CDMA system. Specifically, an analog processing circuit 32 removes the carrier frequency from a signal, such as a CDMA signal received by an antenna 34 and samples the received CDMA signal at a predetermined sampling rate, such as twice the chip rate, to produce corresponding CDMA chip samples. The CDMA chip samples are input into a correlator ASIC 36 in which, according to this specific configuration, all of the components of the post-correlation processing system 12 of the present invention are implemented, including the DSP for implementing, either via hardware or software, the postcorrelation processor 28. Alternatively, if the post-correlation processing system 12 of the present invention is utilized within, for example, a mobile handset environment, the components implemented within the ASIC 36 could alternatively be software implemented. The correlator ASIC 36 outputs generated ontime and non-ontime symbol samples to a DSP buffer 40 in which the early, late and ontime symbol samples E, L, O are stored before being input into a processor 42, such as a Motorola 8102 base station processor, for additional processing including DLL, symbol rate, and chip rate processing as is known in the art.

While the above description is of the preferred embodiment of the present invention, it should be appreciated that the invention may be modified, altered, or varied without deviating from the scope and fair meaning of the following claims.